UNITED STATES PATENT APPLICATION

FOR

PERPENDICULAR VIEW THREE DIMENSIONAL ELECTRONIC PROGRAMMING GUIDE

Inventor: Dan Kikinis

Prepared by:
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026
(408) 720-8598

Attorney's Docket No.: 004688.P020

EXPRESS MAIL CERTIFICATE OF MAILING

Express Mail mailing label number: EL617211084US

Date of Deposit: September 19, 2000

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service Express Mail Post Office to Addressee service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Lisa Kaiser

(Typed or printed name of person mailing paper or fee)

Lisa Kaiser

(Signature of person mailing paper or fee)

September 19, 2000

(Date signed)

PERPENDICULAR VIEW THREE DIMENSIONAL ELECTRONIC PROGRAMMING GUIDE

The present application claims the benefit of the filing date under 35 U.S.C. § 119(e) from U.S. Provisional Patent Application, entitled "Perpendicular-View, Simplified 3-D "Lite" EPG, Serial No. 60/208,883, and also claims priority from the U.S. Provisional Patent Application Serial No. 60/203,128, which was filed on May 8, 2000.

FIELD OF THE INVENTION

The invention relates to the field of creating electronic programming guides (EPGs) for television viewing. More specifically, the invention relates to providing a three dimensional EPG.

BACKGROUND OF THE INVENTION

Electronic Programming Guides (EPGs) allow a viewer to receive information about programs being shown on their cable television system. Currently available EPGs typically display program information, such as which programs are on what channels, and when those programs will be shown. Typically, because there is more information than can be displayed on the television at once, these EPGs scroll past the screen at a specified rate, making

Typically, because there is more information than can be displayed on the television at once, these EPGs scroll past the screen at a specified rate, making navigation frustrating. These EPGs are usually displayed on one channel of a cable television system and do not allow for interactivity.

Newer digital EPGs allow a user to interact using their remote control or other means. These EPGs allow users to scan the program selections and seek out programs which they want to view, and do allow some interaction by the user. However, they are still quite simple, and generally not very attractive.

Digital EPGs also typically require a set top box (STB) to be attached to a user's television. The STB contains hardware and software necessary to process the EPG and the user's input, and also to descramble certain cable channels.

Most EPGs have been simple two-dimensional (2D) text-oriented Electronic Programming Guides. They typically display only one text size and very limited graphics. In general the use of graphics is mostly limited to character-based graphics, capable of drawing boxes, outlines, fields, etc. What is needed is a more interactive and dynamic EPG that can be used on the current generation of STBs.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for displaying an Electronic Programming Guide (EPG) having a reduced graphics hardware requirement and capable of three-dimensional graphics. In one embodiment the method includes displaying a virtual mesh displayed in a perpendicular isometric view. The mesh comprises independent objects situated in a set of planes. The first plane is flat, and subsequent planes are hyperbolic in order to increase legibility. The user can then navigate the mesh, and objects representing television programs can be positioned in the mesh based on predetermined criteria.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which:

Figure 1a illustrates a three dimensional object as generated by a three dimensional graphics pipeline;

Figure 1b illustrates a pixel array;

Figure 1c illustrates a pixel array bound to a parallelogram;

Figure 1d illustrates a pixel array bound to a trapezoid;

Figure 1e illustrates a three dimensional object with a perspective view;

Figure 1f illustrates a three dimensional object with an isometric view;

Figure 2 illustrates an isometric three-dimensional Electronic Programming Guide (EPG) according to one embodiment;

Figure 3 illustrates a category list according to one embodiment;

Figure 4 illustrates the position of an EPG relative to a television screen according to one embodiment; and

Figure 5a illustrates a front view of a perpendicular view Electronic Programming Guide according to one embodiment; and

Figure 5b illustrates a side view of a perpendicular view Electronic Programming Guide according to one embodiment; and

Figure 6 illustrates a system having a readable storage medium with instructions stored thereon according to one embodiment.

DETAILED DESCRIPTION

One embodiment provides a method and apparatus for displaying an Electronic Programming Guide (EPG) with a reduced hardware requirement. In the following description, for purposes of explanation, specific details are set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention.

Commonly assigned U.S. Patent Application Serial No. 09/488,361, filed 01/16/2000, introduced a three-dimensional (3D) EPG which allows for a much more configurable and user friendly interface than previous two dimensional (2D) EPGs.

The invention describe herein is sometimes referred to as 3D EPG "Light." The arrangement and method of displaying the EPG will reduce the hardware requirements of the STB, to further allow current STBs to display the described EPG.

Three Dimensional Graphics Pipeline

In order to display a 3D EPG, a STB typically requires a 3D graphics pipeline. Either a hardware-accelerated 3D graphics pipeline or a software-only 3D graphics pipeline may be used. A hardware-accelerated 3D graphics pipeline has the advantage of faster processing because it has dedicated hardware that

handles calculations that would otherwise be performed by the central processing unit (CPU). A software-only pipeline is typically slower than a hardware-accelerated pipeline because the CPU must handle graphics processing. However, a software pipeline does not require dedicated 3D graphics hardware, and thus is less expensive and can easily be adapted to currently available STBs that do not already have 3D hardware acceleration.

The following is meant to be a short description of a 3D graphics pipeline. The commonly assigned U.S. Patent Application Serial No. 09/488,361, describes 3D accelerator technology in more detail, and is incorporated herein by reference.

Figure 1a illustrates the construction of a 3D object using a 3D pipeline. A 3D graphics pipeline, either hardware or software, operates as a subsystem of a larger computer system that may be part of, for example, a STB. A 3D graphics pipeline creates a 3D object by first receiving data describing the object from the larger system. The data includes coordinates describing a geometric surface, and pixel arrays. A geometric surface 2 is created from polygons 4. For example, in Figure 1a, a baseball bat is constructed from triangles.

A pixel array is in essence a graphic image displayed on a computer system. After the geometric surface is created, the pixel array is bound to the surface. By binding the array to the geometric surface, the 3D pipeline effectively covers the surface with the array. For example, the array corresponding to the object in Figure 1a would contain a bitmapped image resembling wood. The

array would then be bound to the geometric surface, and it would resemble a baseball bat.

Once the image is bound, the 3D pipeline displays the image from a reference point 6. Reference point 6 is the point from where a viewer sees the object. Reference point 6 can be moved to give different views of the object.

Figure 1b is an example of a pixel array. A pixel is essentially a point of light on a display. A display is made up of a grid of pixels, each representing a specific color. The pixels are arranged to create an image on the display. Pixel arrays are image files that are typically rectangular in shape and defined by the number of pixels long by the number of pixels wide. Pixel array 10 is a grid of 9 x 9 pixels. In pixel array 10, there are two colors, black pixels 12 and white pixels 14. Pixel array 10 resembles a house.

As an example, consider that there is a simple geometric surface consisting of a single polygon, a square. A graphics pipeline could bind pixel array 10 to the square. If the square were the same size as the pixel array, the pixel array could be bound to the square by simply placing the pixels within the square. This operation requires minimal processing. Were the square larger or smaller than the pixel array, the pipeline would have to add or remove pixels, or "resize" the pixel array, to approximate the size of the square. Resizing a pixel array still requires only a small amount of processing.

Now, consider binding the pixel array to a rhomboid, such as rhomboid 18 in Figure 1c. If the lines of the rhomboid had the same dimensions as the pixel

array, you could *bit shift* the pixel array. Rhomboidal pixel array 20 is an example of a bit shift of pixel array 10. In rhomboidal pixel array 20, the leftmost column of pixels remains in its place, and the column to the right is shifted up one or more pixels. Each successive column to the right is shifted a certain number of pixels until the rightmost column has been moved. As bitshifting only requires rearrangement of pixels, it too is a relatively minor burden on a graphics subsystem.

If you wanted to bind the pixel array to a trapezoid, such as trapezoid 28 in Figure 1d, you might need to compress some areas of the array, while you might need to expand others. A pixel array is compressed when certain pixels are removed to make the image smaller, and is expanded when pixels are added to make the image larger. As a result, different areas of the image may be compressed or expanded at different rates. Trapezoidal pixel array 30 is an example of binding pixel array 10 to a trapezoid. This type of operation requires a greater amount of processing than either resizing or bitshifting, and can become burdensome on a system that does not have a hardware accelerated graphics pipeline.

Figure 1e is an example of a three dimensional object with a perspective view. When displayed with a perspective view, the lines of a 3D object such as perspective cube 40 will appear to converge upon a vanishing point 42. A perspective view requires processing in addition to the processing necessary to create a 3D object. This is because many of the surfaces of an object with a

perspective view, such as top 44, will be trapezoidal. As explained above, when binding a rectangular pixel array to a trapezoidal surface, extensive processing will be required. Additionally, many of the surfaces in a perspective view will also need to be bitshifted, which further increases the necessary amount of processing.

Figure 1f is an example of a three dimensional object with an isometric view. With an isometric view, a 3D object such as isometric cube 46 appears uniform, and does not seem to disappear at a vanishing point. In other words, parallel lines will always look parallel with an isometric view, while in a perspective view they will appear to converge.

Some rectangular surfaces in an isometric view become rhomboids, such as top 48. As explained above in Figure 1d, binding a pixel array to a rhomboid requires only that a 3D graphics pipeline bitshifts the pixels, a relatively undemanding process. Thus, an EPG that has an isometric view will provide more acceptable performance with a software pipeline. As a result, a software pipeline may be desirable because it can be easily adapted to current set top boxes.

<u>Simplified Three Dimensional Electronic Programming Guide</u>

Figure 2 illustrates an exemplary EPG according to one embodiment. As shown the EPG includes virtual mesh 100 for displaying television programming information. A user may navigate through virtual mesh 100 to find desired

television programs.

Virtual mesh 100 is a polyhedron. A polyhedron is a three dimensional object, such as a cube or a pyramid, that is essentially a solid bounded by polygons. Virtual mesh 100 is also a geometric object (see e.g., Figure 1a, element 2) constructed from independent objects 101, and three planes 102-104. There are relatively few elements to process in virtual mesh 100 when compared to a typical geometric object.

A geometric object, such as geometric surface 2, generally comprises many polygons 4 in order to create a smooth, realistic looking surface. In addition, with a typical scene displayed by a hardware accelerated 3D graphics pipeline contains many geometric objects which all require a significant amount of processing. Even a few geometric objects may comprise several hundred polygons which all require binding bitmap images to their surfaces.

In contrast to most geometric objects, the EPG described below comprises only dozens of surfaces that are mostly stationary. Compared to the typical application which requires a 3D graphics pipeline, the number of elements of the here enumerated EPG is quite small. The simplicity of this EPG is one factor that reduces the amount of processing required by the EPG, and allows it to be implemented on current set top boxes (STBs).

Virtual mesh 100 can be displayed either with an isometric view or in a full 3D perspective view. As explained above, if virtual mesh 100 is displayed in a perspective view, a hardware-accelerated 3-D pipeline may be required.

Displaying virtual mesh 100 with an isometric view may allow the implementation of the EPG on an STB exclusive of a hardware accelerated 3D graphics pipeline. One advantage of the current invention is that new set top boxes may not be required, which may allow this new EPG to be brought to market more quickly and less expensively.

Virtual mesh 100 is constructed of individual mesh lines, which give the mesh its shape and its structure. As shown in Figure 2, virtual mesh 100 has the shape of a cube, but it is clear that virtual mesh 100 may also be in the shape of other polyhedrons. The mesh lines are arranged as to create a set of columns and rows, in which television programming information can be displayed.

As shown in Figure 2, virtual mesh 100 is a group of lines displayed on a video screen configured so that objects 101 may be displayed within the mesh. In one embodiment, objects 101 represent individual television programs. Independent objects 101 are arranged as to make navigation of the mesh simple, and to bring to the user's attention those programs in which they may be interested.

Virtual mesh 100 is further a virtual 3-D view of three parallel planes 102, 103 and 104 housing at least one independent object 101 positioned in virtual mesh 100. Virtual mesh 100 is further aligned along three axes, the x-axis, the y-axis and the z-axis. As shown in Figure 2, planes 102, 103, and 104 are aligned along the z-axis.

In one embodiment, planes 102-104 are rectangular and parallel. The first

plane 102 creates the face of mesh 100, the second plane 103 bisects mesh 100, and the third plane 104 creates the backing of mesh 100. The three planes house objects 101 so as to give the appearance of a table of television listings.

Each plane may have its own color and form. For example, plane 102 can be assigned the color red, and thus its mesh lines are red. Plane 103 can be assigned the color blue, and thus its mesh lines are blue, and plane 104 can be assigned the color green, which means its mesh lines are green.

In one embodiment, objects 101 are not a part of virtual mesh 100; they are merely situated within the mesh 100. Each of the program elements 101 is a fully independent object that may be manipulated by the EPG independent of virtual mesh 100. Objects 101 further comprise one or more interactive surfaces. Interactive surfaces may be acted upon by the user to create a desired result, such as selecting a new channel.

Objects 101 may be arranged in planes 102, 103, 104 according to user selected categories. In one embodiment, the three planes shown represent levels of preference, and specifically represent the categories "preferred," "neutral," and "don't prefer." However, any alternative number of categories greater than one could be used. In Figure 2, the first plane 102 represents the "preferred" category, the second plane 103 represents the "neutral" category and the third plane 104 represents the "don't prefer" category.

Although one embodiment shows planes 102, 103 and 104 that are parallel with a rectangular shape, and objects 101 that are rectangular, in other

embodiments the program listings could be displayed in various different forms. An object 101 could be a pictogram that graphically displays the type of programming represented by that object. For example, the program listing of a baseball game could appear in a virtual 3-D shape of a baseball bat, or a baseball (not shown). Additionally, the planes need not be parallel and aligned along the z-axis. Any variety of shapes, sizes, and alignments of listings and listing planes could be created within the scope of the invention.

The top plane of virtual mesh 100, formed by the x and y axes of the three planes is a Time Plane. It has time numerals 105 a through n, the first two of which are shown as "9:00" and "9:30". Time lines 106a through 106n extend along the z axis from front to back, as seen in Figure 2, across planes 102, 103, and 104. The numbers and the time lines may also be a distinctive color, such as, for example, yellow or gold.

Mesh rows 108a through 108n represent channels, and columns 110a through 110n represent times. Names identifying channels are listed in the first column 110a. The rows corresponding to a specific channel listed in column 110a further list the programs on that channel at different times. For example, row 108b contains listings for the channel "ABC" 111.

Virtual mesh 100 can contain all available channels from a service provider or can contain a subset of channels selected by interest group or by certain other criteria. The selection criteria may be either user-selectable or pre-selected by the provider.

A user may navigate the mesh using a remote control device attached to a set-top box (not shown). A user navigates mesh 100 by moving a cursor or highlight bar throughout the mesh. For example, in Figure 2, row 108b is highlighted by highlight bar 112, which indicates to the user that they have currently selected that channel. In Figure 2, the selected channel is ABC 111. Because row 108b is highlighted, a semi-transparent colored band extends from the front row to the back to indicate the location of the elements.

The cursor 114 is currently highlighting element 116, which is an object that represents the user's currently selected program, the news. In one embodiment, because the news 116 is currently selected, the full title of the program "ABC Evening News" is displayed on sign 120. Sign 120 is placed on top of the mesh and displays details of the selected program. The user may move cursor 114 throughout the mesh 100 to select and preview different television programs. In addition to showing the selected element on table 120, a further enhancement in one embodiment could pull that show to the front and display it hovering in front of the mesh 100.

In the example illustration of Figure 2, the news 114 is in the first plane 102 because the user has designated newscasts as a preferred television program. Based on selected interest categories, the programs matching certain criteria are displayed in the first plane 102, indicating shows of the highest interest, the second plane 103 indicating shows of neutral interest, or displayed in the third plane 104, indicating shows of the lowest interest. Although planes 102, 103, and

104 are described above as representing level of interest, it is clear than a variety of criterias may be used to determine which programs are listed in which planes.

Furthermore, in the example shown in Figure 2, Johnny Sitcom 122 is set back into the second plane 103, "neutral," because the user has given sitcoms a "neutral" preference setting. Womantalk 124 is set back into the third plane 104, or "not preferred."

The user views virtual mesh 100 from the front. Program listings in the front plane 102 appear to be at the front of the screen. Program listings in the middle plane 103 appear to be set back from the front of the screen, and program listings in the back plane 104 appear to be set further back than program listings in plane 103. In one embodiment, the programs that the user wishes to view are displayed more prominently, so that the user's attention will be immediately drawn to them. This increases the ease of use of the EPG and creates an attractive visual effect.

Although this example of one embodiment shows three planes, the EPG may contain any number of planes, such as only two, representing interested and not interested, or four or five planes, for varying levels of interest. In yet another embodiment, multiple users on the same set-top box may have different preferential profiles and may accordingly assign different categories to different preference levels.

The reduced number of elements that comprise the above described EPG helps to reduce the hardware requirements. Compared to most geometric

objects processed by 3D pipelines, virtual mesh 100 and independent objects 101 are quite simple, and therefore will not require a great deal of processing to display. Additionally, the number of objects that comprise the EPG is also comparatively few, and thus there is less processing required.

Figure 3 illustrates an example of a category list where all categories of the various available programs are shown. List 200 has two columns. Column 201 lists categories and column 202 shows the interest level assignment of each category. Either a user-selectable or preassigned value is assigned to each category. These values are then used by the system to tie the shows falling into each category to the correct planes 102 through 104.

In Figure 3, for example, sports programs are given an interest level 1, and would therefore be in the first plane 102. Likewise, sitcoms are assigned interest level 2, and would be set in the second plane 103. And, late night shows are given interest level 3, and would be set in the third plane, 104.

Multiple customized tables can be programmed for different users, so when a user identifies himself, the EPG rearranges shows within the mesh accordingly. By pushing back certain shows and also making their entries visibly smaller than those in the foreground, shows of lesser interest are de-emphasized.

Figure 4 shows virtual mesh 100 as seen on a television screen. In one embodiment, the angle between the top line and the perpendicular, as viewed on a 2-D screen, is between 90 and 97 degrees. The embodiment shown in Figure 4 is exemplary. Other arrangements may be chosen without affecting the

functionality of the EPG.

Hyperbolic Correction

Figures 5a-5b represent a further embodiment of the EPG discussed in Figures 1-3. Figures 5a-5b represent a perpendicular view of virtual mesh 100, with additions to improve legibility. As explained above when using a perspective view, there is additional processing required. However, with a perpendicular view, the amount of processing is reduced because any images bound to a surface that is perpendicular to the line of sight can be displayed with minimal manipulation. If an image were bound to a surface such as top 44 of Figure 1e, a 3D graphics pipeline would typically have to resize the image and distort it. However, if you were to bind an image to face 49 of Figure 1f, the most manipulation required would be resizing the image, if any manipulation were required at all.

A perpendicular isometric view also requires less processing than a standard isometric view as shown in Figure 1f. With a perpendicular view, fewer surfaces such as top 48 will be visible, and thus fewer images will have to be bitshifted in order to be bound to such surfaces. The location of certain surfaces will still require manipulation of pixel arrays bound to them, however the amount of processing is still greatly reduced.

A hardware three dimensional (3D) graphics pipeline can process a perspective view with acceptable speed, but most STBs on the market are not equipped with a hardware-accelerated 3D graphics pipeline. Thus, a

perpendicular view reduces the hardware requirements and can be adapted to current STBs using a software pipeline.

The perpendicular view of this invention is an isometric view. However, the manipulation of certain elements of the virtual mesh gives the illusion of a perspective view. Viewers generally view a television, and thus the EPG from above, and as a result, in Figures 5a and 5b there is the illusion a focal point located behind the EPG.

The viewer views the EPG from above, and thus it may be difficult to see the listings on the bottom if the planes are parallel. There are several solutions to this problem. One is to reduce the distance between the planes, but this may eventually destroy the three dimensional effect. Another solution is to make a non-linear correction so that various program description surfaces are placed differently, based on the nature of the viewer's perspective.

One embodiment has all planes other than the front plane of the EPG being geometric surfaces. As shown in Figures 5a-5b, the geometric surfaces are hyperbolic planes. A hyperbolic plane reduces the distance between the first plane and the second plane near the bottom of the EPG, where a user may have difficulty seeing objects set back into the second plane. In other words, in a two plane EPG, the second plane will be a surface that conforms to a hyperbola such as $y=A+1/(Bx+C)$. The hyperbolic correction only affects the y-axis. Were an EPG to be displayed with a slight perspective view, corrections would be needed

in both the x-axis and the y-axis, and there would be a corresponding increase in processing required.

Figure 5a illustrates an exemplary two plane EPG with a hyperbolic correction in the second plane. Virtual focal point 405 is located above virtual mesh 400. The viewer's line of sight is on the level of focal point 405 and is perpendicular to the face of virtual mesh 400. Flight lines 410 are shown to illustrate the illusion of perspective of the EPG, but are not present in this embodiment. Flight lines 410 converge upon focal point 405.

As illustrated in Figure 4a, virtual mesh 400 is constructed from individual independent objects 412, the nature of which are explained above. Virtual mesh 400 is further constructed from two planes, a front plane 413 and a back plane 414. The front plane 413 is flat, all independent objects 412 situated within front plane 413 appear to be the same distance from the viewer. The back plane 414 is a hyperbolic plane, it is a curved geometric surface that follows the path of a function defined by $y=A+1/(Bx+C)$.

As with the planes in Figure 1, the front plane 413 may represent "preferred" listings and the back plane 414 may represent "not preferred." It is also clear that any type of classification may be used in place of levels of preference, and that additional planes may be added.

Objects 412 do not conform to the shape of back plane 414. Instead, objects 412 are parallel to front plane 413 and use back plane 414 as a guide in

determining their distance from front plane 413. In one embodiment, one of three algorithms determines the object 412's position:

1. Compute the new positions of the four corner vertices and take an average as a final position.
2. Compute the new positions of two corners and take the maximum or minimum as a final position.
3. Compute the position of the rectangle's center as a final position.

Virtual mesh 400 is divided into two columns 415a-b representing times during which television programs will be shown. Time displays 420 indicate to the user the time to which each column corresponds. Virtual mesh 400 is also divided into several rows 425a-h which represent television stations. Columns 415 and rows 425 define individual cells. Each cell is occupied by at least one independent object 412. Each object 412 represents at least one "interactive surface," which is a region upon which a user may act to create a desired result. In one embodiment, objects 412 represent television programs. So, for example, a user could select a television program, which is an interactive surface, and have the EPG direct the television to tune in that station. Objects 412 located in rows 425c, 425e are set into the back plane, and thus conform to the hyperbolic surface 414.

Figure 5b illustrates a side view of the two plane EPG illustrated in Figure 5a. Figure 5b makes the presence of the hyperbolic plane 414 much clearer. The

objects 412 located in rows 425c and 425e are set back along back plane 414 by distances 435a-h.

The geometric surfaces described above may be specified by any one of a number of different curves, such as parabolas or sections of ellipses. Also, the user's position may change, and the EPG may be adjusted to compensate for their new reference point. It is also possible to display this EPG with a perspective perpendicular view when utilizing a STB of sufficient processing power.

The method of displaying an EPG in three dimensions, as described above, can be stored in the memory of a computer system (e.g., set top box, video recorders, etc.) as a set of instructions to be executed, as shown by way of example in Figure 6. In addition, the instructions to display an EPG in three dimensions as described above could alternatively be stored on other forms of machine-readable medium, including magnetic and optical disks. For example, the method of the present invention could be stored on machine-readable mediums, such as magnetic disks or optical disks, which are accessible via a disk drive (or computer-readable medium drive). Further, the instructions can be downloaded into a computing device over a data network in a form of compiled and linked version.

Alternatively, the logic to perform the methods as discussed above, could be implemented in additional computer and/or machine readable mediums, such as discrete hardware components as large-scale integrated circuits (LSI's),

application-specific integrated circuits (ASIC's), firmware such as electrically erasable programmable read-only memory (EEPROM's); and electrical, optical, acoustical and other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

The embodiments above have been described in sufficient detail with a certain degree of particularity. It is understood to those skilled in the art that the present disclosure of embodiments has been made by way of examples only and that numerous changes in the arrangement and combination of parts may be resorted without departing from the spirit and scope of the embodiments as claimed. Accordingly, the scope is defined by the appended claims rather than the forgoing descriptions of embodiments.